



PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Algirdas Avizienis	Group
Serial No.:	09/886,959	Art
Filed:	June 20, 2001	Unit:
Title:	"SELF-TESTING AND -REPAIRING FAULT-TOLERANCE INFRASTRUC- TURE FOR COMPUTER SYSTEMS"	2613
Our docket:	xAAA-02	Examiner Bryce P. Bonzo

DECLARATION OF JEAN-CLAUDE LAPRIE, Doctor es-Science

Hon. Asst. Commissioner for Patents
P. O. Box 1450
Arlington VA 22313-1450

Sir:

I, Jean-Claude Laprie, declare as follows.

1. I am a Directeur de Recherche (Director of Research) at the French National Organization for Scientific Research ("CNRS") in Toulouse — and particularly its Laboratory for System Analysis and Architectures ("LAAS").
2. I am a specialist in computer reliability, dependability, fault tolerance, and security.

JCL

3. I joined LAAS-CNRS in 1968. Here I founded the research group on fault tolerance and dependable computing in 1975, and directed that group until I became the director of LAAS in 1997.
4. My research has focused on dependable computing since 1973, and especially on fault tolerance and on dependability evaluation — subjects on which I have authored and coauthored more than a hundred papers, while also coauthoring or editing several books.
5. I have no monetary interest in the present patent application, and no financial interest in the affairs of the Applicant, Dr. Avizienis — and no other reason for favorable bias toward him that would prejudice my statements in this declaration. In years past, I coauthored some papers with him; this past collaboration in no way clouds my objectivity toward this subject matter in general, or his inventions specifically. My participation in this proceeding by declaration is without compensation, as part of the general exchange of personnel evaluations, literature critique etc. — among academic colleagues — again without any implication of bias.
6. I have been very active in the formulation of the basic concepts of dependability and the associated terminology; the views developed being widely adopted by the scientific community. My activities have included many collaborations with industry, culminating in the 1992 foundation of the Laboratory for Dependability Engineering ("LIS"), a joint academia-industry laboratory, that I directed until 1996.

7. I have also been very active in the international community, including the chairmanship of the IEEE Computer Society Technical Committee on Fault-Tolerant Computing in 1984-1985, of the International Federation for Information Processing ("IFIP") Work Group 10.4 on Dependable Computing and Fault Tolerance from 1986 to 1995, and of IFIP Technical Conference 10 on Computer System Technology from 1997 to 2001.
8. I am currently a vice-president of IFIP, and the representative of France in the IFIP General Assembly.
9. My honors include receipt of the IFIP "Silver Core" in 1992, the Silver Medal of the French Scientific Research in 1993, and the National Merit Medal in December 2002.
10. In summary, I am an internationally recognized expert and authority in the field of computing reliability.
11. In 2000 in New York City, at the Dependable Systems and Networks Conference, when Dr. Avizienis presented his technical paper on the system and methods that are the subject of this patent application, I was impressed with its innovativeness and its distinctive inventiveness — and I was one of several experts in this field who told him so.
12. As I understand it, Dr. Avizienis's patent application contains dozens of claims, and all but one are rejected. I am told that the grounds of rejection differ, for the many different claims, but do have several common points that are repeated.

13. In order to contribute to the dialogue, I have studied just a few claims that I am told are representative of all the claims, and whose rejections I am told are representative of all the rejections. I have also studied those rejections.
14. Based upon my expertise in this field as described above, I shall comment on the rejections of those representative claims. The representative claims are: claims 3, 18, 33, and 46.
15. I understand that one of these claims is "independent"
i. e., stands alone without referring to other claims. That is claim 33.
16. I also understand that the others of these claims are "de-
pendent" — they cannot stand alone, but necessarily in-
corporate by reference some features that are recited in
related independent claims. Therefore in discussing the
dependent claims (claims 3, 18 and 46) I shall also comment
on their associated independent claims (those are claims 1,
13 and 42).
17. CLAIM 3 — This claim, with its related independent claim 1,
reads thus:
- a 1. Apparatus for deterring failure of a computing system; said apparatus comprising:
b a hardware network of components, having substantially no software and substantially no
c firmware except programs held in an unalterable read-only memory;
d terminals of the network for connection to such system; and
e fabrication-preprogrammed hardware circuits of the network for guarding such system from
f such failure.
- a 3. The apparatus of claim 1, wherein:
b the network is an infrastructure which is generic in that it can accommodate any such system
c that can issue an error message and handle a recovery command.

TEL

18. On one hand, I understand that the phrase "computing system" refers to the computer(s) to be protected against failure. On the other hand, I understand that the terms "apparatus", "hardware network of components", "terminals" and "circuits" all refer to Dr. Avizienis's infrastructure invention, or parts of it.
19. I understand that the patent Examiner has cited against the representative claims (a) a patent issued to a man named Best, in combination with (b) Dr. Avizienis's own 1985 paper which discusses the Design Diversity Experiment ("DEDIX"). I have read and carefully considered the portions of both those documents on which the Examiner's rejections rely.
20. I see that the Examiner's discussion (in his "Official Action") of claim 3 says:

Best does not disclose, while Avizienis [1985] teaches:

at least one of the network terminals is connected to receive one error signal generated by such system in event of incipient failure of such a system (page 1498 describes how the Decision and Executive layer receives exceptions from the version layer indicating errors within the instance running that particular version of the software);

at least one of the network terminals is connected to provide one recovery signal to such system upon receipt of the error signal (page 1498 describe[s] the use of the local executive processing faults and providing [a] solution to the problem); and

the apparatus further comprises means for automatically responding to the at least one error signal by generating the at least one recovery signal for guarding all such system against failure (page 1498 discloses the local and global executives at differing levels providing commands to the version which prevent[s] failure).

21. That passage specifically addresses claim 3, but the Examiner's discussion continues with the following paragraph. (I understand that the entire paragraph below is thereafter also replicated verbatim with regard to some twenty other rejections.)

Best as shown above discloses a hardware based system to monitor plural processing/computing channels for errors in a separate computing system (column 3, lines 15-29); column 4, lines 14-25). Best further discloses that his system is [sic, has?] application to all types of digital systems (column 6, lines 29-61) and further are [sic] ... applicable at any level (column 6, lines 42-47). Avizienis teaches the use of the DEDIX

OC1

distributed data processing system, and more importantly that is [s/c, it?] may implemented [s/c] in single computer and multiple computer [s/c] acting in concert across a network (page 1497, column 2). Avizienis further describes the fault handling system as being separated into separate distinct layers (page 1497, column 2). Avizienis further describes the needs as an architectural need: hardware voting and consistency checking (page 1496, column 1). From these passages one of ordinary skill concludes that Avizienis has expressed a need for a hardware support structure to manage the voting he describes, and that it can be implemented in multiple distribution styles. Best provides a clear intend [s/c, intent?] to be used in hardened voting schemes with processors which may go astray and must be corrected timely (specifically avionics, which as a side note Avizienis produces later papers on the need for such system in space craft, as specialization of avionics). Therefor [s/c] it would have been obvious to one of ordinary skill in the art at the time of the invention to implement the fault tolerant portions of the layered DEDIX system of Avizienis with the hardware fault locating and handling system of Best, thereby creating a stronger N-version software system.

22. Thus the Examiner appears to propose that a person skilled in the computer field would find the apparatus of Dr. Avizienis's claim 3 "obvious" as a combination of certain features of Best with certain features of Dr. Avizienis's own 1985 paper.
23. After pondering the Examiner's two passages above, and carefully studying the cited parts of those two earlier documents, I sincerely cannot agree. Here is why:
24. First, neither of the two reference documents even satisfies the recitation of the underlying independent claim 1: "a hardware network of components, having substantially no software" Best makes absolutely clear that his product is "under software control" and has a host processor (though it is vaguely defined, it evidently runs software); and DEDIX is entirely software.
25. Therefore, as a completely independent expert in this field, with apologies I find it difficult to understand how there can be any argument about this extremely simple technical point.

JCL

26. Second, the Examiner's notes reveal several misunderstandings of the character of the systems in the two references, vis-a-vis the claim-3 language (emphasis added): "the network is an infrastructure which is generic in that it can accommodate any such system that can issue an error message and handle a recovery command." This is a very simple and short clause, but the Examiner talks all around it (see paragraphs 20 and 21 above) without ever confronting it directly.
27. As I read the Avizienis 1985 discussion of the DEDIX, that experiment was set up to deal with a very specific class of software modules that were custom-designed for the experiment. They were specially fitted with very unusual cross-check features, to enable the N-version software comparisons in the experiment. Similarly, many other kinds of software modules would not be compatible with the DEDIX at all.
28. The cross-check provisions of the DEDIX required that each participating program of the N versions contained specifically formatted "cross-check vectors" ("cc-vectors") that the DEDIX could recognize. Furthermore they had to be at particular points in the program, where data were comparable as between versions. A program lacking these cc-vectors — at the right locations — could not run properly in the DEDIX. Thus it is quite wrong to call the DEDIX "generic in that it can accommodate any such system that can issue an error message and handle a recovery command." Note too that a cc-vector was not an "error message".
29. Best likewise is controlling a set of circuit modules that are extremely distinctive, and the control signals from

JCL

their associated software are custom-designed to operate those particular circuit modules. Thus neither of the two cited references answers to the simple, direct language of claim 3.

30. Paragraphs 24 through 29 above give two reasons to question the purported "obviousness" of claim 3 in view of the Avizi-enis/Best combination: neither reference meets the underlying claim-1 limitations, and neither reference meets the plain claim-3 limitations. Although I believe that each one of these reasons alone is fatal to the Examiner's argument, I will discuss below (in paragraphs 31 through 40) yet a third reason that is more directly responsive to the Examiner's sophisticated arguments: the documents do not inspire a person who knows this field to make the combination.
31. Third, as I have indicated just above, a person who is skilled in this field and looks at the two documents will not likely be moved to combine any of their features.
32. One document describes a trivial hardware circuit for manufacture, practical application and ongoing use. The other describes a distributed operating system — closely supervised by many scientists working in parallel — for a one-time academic experiment about executing and debugging multiple-version software systems.
33. I do understand, and I will grant, that under some circumstances a patent examiner normally may be at liberty to disregard such practicalities and basics. In this situation, however, the issue seems to be what a real person skilled in

JCL

the field would perceive or realize about possible "combinations" — and to that issue, I can testify with confidence.

34. In truth, in the real world, such a person's thinking is illuminated by the relationships that suggest themselves from reading the two documents. In this case, virtually no relevant relationships suggest themselves. The concepts, and especially the documents, are just too disparate.
35. In fact, even upon being told that it is suggested to somehow combine the two, someone skilled in this field must immediately wonder why.
36. The purported linkage is just not real, and in fact it is almost incomprehensible to me how the DEDIX experiment could be implemented in hardware, or Best could be expanded into an N-version software experiment, or why.
37. In an early section of Dr. Avizienis's 1985 paper — not discussing the DEDIX — he does suggest that a general-purpose computer could be improved by including two additional instructions in the instruction set: "take majority vote", and "cross-check" data (at the cross-check points). Such proposed instructions might be regarded as "hardware support" for software voting, but the Examiner's argument that Avizienis suggested "hardware voting" is flatly incorrect. The Avizienis '85 paper suggests neither (a) hardware voting nor (b) combination of an N-version software experiment with hardware voting.
38. Still as to the third reason for believing that it would not be obvious to combine Best with Avizienis '85: although it

would not occur to a person in this field to combine them, ironically the insertion of Best into the DEDIX also would be superfluous. The DEDIX already has voting functions.

39. I see no reason to believe that Best's voting hardware, with associated software modules to run it, would be any better than the all-software voting provisions already present in the DEDIX. Therefore even if it did, for some incomprehensible reason, occur to a skilled computer engineer to combine the two ideas, that person would immediately reject the idea.

40. Furthermore as a practical matter the importation of Best's hardware, or its conceptual equivalent, into the multiple-minicomputer network running DEDIX would pose major practical problems. Not only would an instance of Best's hardware have to be electronically linked with each one of the minicomputers, but in addition Best's control software would have to be somehow integrated into the existing DEDIX software — and all to no particular avail, since the hardware implementation (as I have stated above) offers no benefit.

41. CLAIM 18 — This claim, with its independent claim 13, reads as follows. (Please note the ellipsis ". . ." at line c — where several lines have been omitted because they are not related to the Best patent or the Avizienis 1985 paper.)

a 13. Apparatus for deterring failure of an entire computing system, wherein the computing system
b optionally includes plural mutually redundant modules; said apparatus comprising:
c a network of components having terminals for connection to such system, . . . ; and
d circuits of the network for operating programs to guard such entire system from such failure;
e the circuits comprising portions for identifying such failure of any of the circuits and
f correcting for the identified such failure.

a 18. The apparatus of claim 13, wherein:
b said circuits receive from such system error messages warning of incipient such failure, and
c issue recovery commands to such system.

JCL

42. The assertion by the Examiner as to claim 18 is:

Best does not disclose, while Avizienis teaches:

said circuits receive from such system error messages warning of incipient failure, and issue recovery commands to such system (page 1498 discloses specific application and OS error recovery commands handled by the Local executive, while Best discloses substituting the correct data onto the channel).

This comment is followed by yet another copy of the longer paragraph quoted previously.

43. First, in my professional opinion neither Best nor the 1985 Avizienis paper answers to the language of the independent claim, claim 13. In support of this statement, I note:

44. Best does not deter or guard against failure of an "entire" computing system, and indeed never even discloses the basic character of the computing system — it is at most only implicitly present.

45. Since Best presents almost no information about the computing system to be protected, certainly he does not suggest that his modest invention can protect that (or any other) "entire" computing system. His invention relates to only those small parts (the "communication channels") of a computer network that pass messages, nothing more.

46. As to Dr. Avizienis's discussion of the DEDIX experiment, it was not the particular purpose of that academic experiment to guard against failure — but rather to monitor and study failure.

47. For the purposes of academicians, the experiment evidently was run unattended for relatively brief periods (e. g. overnight or on a weekend), and would collect errors for study

Jcl

and analysis. It would also allow the supervising experimenters to debug the many ("N") program versions involved, individually.

48. Thus any possible effect which the experiment may have had in "detracting failure" of its individual program versions was incidental to its primary functioning — namely to learn what could be learned about the parallel operation of multiple program versions. In other words in the DEDIX the deterrence of individual-version failure is a rather semantic question, revolving around the relative importance of primary purpose vs. the relative importance of incidental effects.
49. For present purposes, however, all of this is a moot point — because the language of claims 13 and 18 specifies "detracting failure of an entire computing system", and this the DEDIX certainly did not do. In no way could the all-software DEDIX prevent the entire system, consisting of some twenty minicomputers, from failing.
50. Since the specifics of claim 13 are understood to carry over to claim 18 as well, then claim 18 is likewise unable to read on Best, or DEDIX, or any combination of the two.
(Once again I find the idea of combining these two documents very unobvious to a person skilled in the computer field.)

TCL

51. CLAIM 33 — This claim is independent:

a 33. Apparatus for deterring failure of a computing system that is substantially exclusively made
b of commercial, off-the-shelf components and that has at least one hardware subsystem for
c generating an error message of the system about incipient failure; said apparatus comprising:
d a network of components having terminals for connection to such system; and
e circuits of the network for operating programs to guard such system from failure;
f the circuits comprising portions for reacting to such error message of such hardware
g subsystem.

52. As to this claim, the Examiner again asserts (questionably) that Best's apparatus deters failure "of a computing system", and includes circuits "for operating programs to guard such system from failure". The Examiner then comments further:

Best does not disclose, while Avizienis teaches:

Apparatus [s/c, "a computing system"] that is substantially exclusively made of commercial, off-the-shelf components and that has at least one hardware subsystem for generating an error message of the system about incipient failure (page 1498 describes the Version layer reporting errors and receiving decisions results), the portions for reacting to such error messages of such hardware system (1498 discloses the Local and global executives performing these functions).

53. I disregard the apparent typographical error, in which the Examiner at first seems to mistakenly attribute the all-COTS construction and the hardware subsystem to the "apparatus" rather than the claimed "computing system". I think it is clear what he meant.

54. Here too, however, it appears to me that the analysis is badly off target. The truth is that the version layer of the DEDIX, as described by Avizienis, does not have "at least one hardware subsystem for generating an error message . . . about incipient failure". My statement here is correct for the "version layer" in Fig. 2 of Avizienis as well as the general "version J" in Fig. 1.

55. The DEDIX is an all-software experimental arrangement — using essentially standard minicomputers operated by its aca-

JCL

demic participants. All of the N versions in the "version layer" of the DEDIX are likewise all software. There is no ASIC for generating error messages. Therefore it does not make sense to me to describe any part of these general-purpose minicomputers, or any part of the software running in them, as "a hardware subsystem".

56. In my professional opinion it is improper to strain the normal definitions of computer-system elements in this way, in an effort to "prove" that a person of ordinary skill in the field would find it obvious to combine teachings of two references. The definition implied here is overbroad and therefore unfair.

57. I think such argumentation departs unreasonably from a patent Examiner's prerogative to adopt a broad but fair reading of claim language. In real life, a person skilled in the computer field would not recognize, or acquiesce in, such strained definitions of ordinary elements.

58. CLAIM 46 — This claim, with its independent claim 42, reads thus (here too, at the ellipsis "... " in line g, several irrelevant lines have been omitted):

a 42. Apparatus for deterring failure of an entire computing system that is distinct from the
b apparatus and that has plural generally parallel computing channels and has at least one
c application-data input module, and at least one processor for running an application program; said
d apparatus comprising:
e a network of components having terminals for connection to such system; and
f circuits of the network for operating programs to guard such entire system from such failure,
g ... ;
h the circuits comprising portions for comparing computational results from such parallel
i channels.

a 46. The apparatus of claim 42, wherein:
b the network is an infrastructure which is generic in that it can accommodate any such system
c that can issue an error message and computational results, and handle a recovery command.

59. I have commented above on the "entire computing system" concept. That previous comment applies also to claim 42.

60. The patent Examiner makes short work of claim 46, but to me his statements about this claim are simply wrong. He says:

Best does not explicitly disclose, while Avizienis teaches:

the network is an infrastructure which is generic in that it can accommodate any such system that can issue an error message and handle recovery command (page 1498; DEDIX hides all the fault processing from the version layer, and it is the decision and executive layer which performs this functionality, making it generic).

61. First regarding the language of independent claim 42, as I have pointed out above, the DEDIX did not and could not deter failure of an "entire" system. (At the very most it could incidentally deter failure of some subcomponents of its multiple versions, and not as its major purpose — and even then only for short time intervals between supervisory interventions by its experiment operators.)

62. Second, regarding the language of dependent claim 46, as I have pointed out above, neither Best nor the DEDIX could accommodate "any" system that could issue an error message and handle a recovery command; rather, both of these reference inventions were able to service only very narrowly defined and prepared (basically "custom") systems.

63. Actually I think the DEDIX network was not truly an infrastructure at all. In my view an infrastructure provides to a computing system all the necessities for operation, including power, active isolation from potentially hostile physical environments, and (particularly in the case of the present invention) fault tolerance — i. e., ability to continue functioning in the presence of faults, including for instance hacker attacks. The DEDIX was not an infrastruc-

jc)

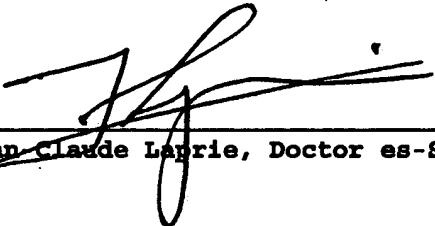
ture but rather in the nature of an intermediate-level operating system (or operating subsystem), being built upon a basic layer of Unix as shown in the diagrams.

64. Third, the Examiner's brief statement about claim 46 seems to go through the motions of a logical process; but, as I see it upon careful reflection, the conclusions do not follow from the premises — and, what is more, are not correct.
65. To be specific, the fact that fault processing is hidden from the version layer proves virtually nothing. In particular it does not imply that the DEDIX network can accommodate any system (or any system which can issue error messages and handle recovery commands).
66. Similarly the fact that the decision and executive layer performed the hiding or provided error messages does not prove that the DEDIX was "generic" in the sense defined in claim 46.
67. Therefore it strikes me as reckless, and also definitely not correct, to state with authority that the DEDIX "is an infrastructure which is generic in that it can accommodate any such system".
68. After his brief paragraph, quoted just above, about claim 46, the Examiner again replicates his long paragraph about combining teachings of the two references. For reasons stated earlier, I object to this combining. I find the idea of constructing such a combination bizarre.

69. My remarks about "all hardware" construction, and about protecting an all-COTS computing system, and about generically protecting "any system", and about receiving an error message and returning a recovery command, and about protecting an "entire system" of course apply to other claims that recite substantially the same features, respectively.

All statements herein made of my own knowledge are true; all statements made on information and belief I believe to be true. I understand that willful false statements and the like herein are punishable by fine or imprisonment, or both (18 U.S.C. 1001) and may jeopardize the validity of the subject application or any patent issued thereon.

May 29, 2006
date



Jean-Claude Laprie, Doctor es-Science